COMPUTER SCIENCE

A. K. Watson Hall, 203.432.1246
http://cpsc.yale.edu
M.S., M.Phil., Ph.D.

Chair
Zhong Shao

Director of Graduate Studies
Vladimir Rokhlin (108 AKW, 203.432.1283, vladimir.rokhlin@yale.edu)

Professors Dana Angluin, James Aspnes, Dirk Bergemann,* Ronald Coifman,* Julie Dorsey, Stanley Eisenstat, Joan Feigenbaum, Michael Fischer, David Gelernter, Mark Gerstein,* John Lafferty,* Rajit Manohar,* Drew McDermott (Emeritus), Dragomir Radev, Vladimir Rokhlin,† Holly Rushmeier, Brian Scassellati, Martin Schultz (Emeritus), Zhong Shao, Avi Silberschatz, Daniel Spielman, Leandros Tassiulas,* Y. Richard Yang, Steven Zucker†

Associate Professor Mahesh Balakrishnan

Assistant Professors Wenjun Hu,* Julian Jara-Ettinger,* Amin Karbasi,* Smita Krishnaswamy,* Sahit Negahban,* Ruzica Piskac, Mariana Raykova, Jakub Szefer,* Marynel Vázquez

Senior Lecturer Stephen Slade

Lecturers Benedict Brown, James Glenn, Kyle Jensen,* Scott Petersen, Brad Rosen, Andrew Sherman, Xiyin Tang [Sp]

* A secondary appointment with primary affiliation in another department or school.
† A joint appointment with another department.

FIELDS OF STUDY
Algorithms and computational complexity, artificial intelligence, data networking, databases, graphics, machine learning, programming languages, robotics, scientific computing, security and privacy, and systems.

RESEARCH FACILITIES
The department operates a high-bandwidth, local-area computer network based mainly on distributed workstations and servers, with connections to worldwide networks. Workstations include Dell dual-processor PCs (running Linux or Windows/XP). Laboratory contains specialized equipment for graphics, vision, and robotics research. Various printers, including color printers, as well as image scanners, are also available. The primary educational facility consists of thirty-seven PC workstations supported by a large Intel PC server. This facility is used for courses and unsponsored research by Computer Science majors and first-year graduate students. Access to computing, through both the workstations and remote login facilities, is available to everyone in the department.

SPECIAL ADMISSIONS REQUIREMENTS
Applicants for admission should have strong preparation in mathematics, engineering, or science. They should be competent in programming but need no computer science beyond that basic level. The GRE General Test is required.

SPECIAL REQUIREMENTS FOR THE PH.D. DEGREE
There is no foreign language requirement. To be admitted to candidacy, a student must (1) pass ten courses (including CPSC 690 and CPSC 691) with at least two grades of Honors, the remainder at least High Pass, including three advanced courses in an area of specialization; (2) take six advanced courses in areas of general computer science; (3) successfully complete a research project in CPSC 690, CPSC 691, and submit a written report on it to the faculty; (4) pass a qualifying examination in an area of specialization; (5) be accepted as a thesis student by a regular department faculty member; (6) serve as a teaching assistant for two terms at a TF level 10; and (7) submit a written dissertation prospectus, with a tentative title for the dissertation. To satisfy the distribution requirement (requirement 2 above), the student must take one course in programming languages or systems, one programming-intensive course, two theory courses, and two in application areas. In order to gain teaching experience, all graduate students are required to serve as teaching assistants for two terms during their first three years of study. All requirements for admission to candidacy must be completed prior to the end of the third year. In addition to all other requirements, students must successfully complete CPSC 991, Ethical Conduct of Research, prior to the end of their first year of study. This requirement must be met prior to registering for a second year of study.

MASTER’S DEGREES
M.Phil. See Degree Requirements under Policies and Regulations.

M.S. (en route to the Ph.D.) To qualify for the M.S., the student must pass eight courses at the 500 level or above from an approved list. An average grade of at least High Pass is required, with at least one grade of Honors.
Terminal Master’s Degree Program Students may also be admitted to a terminal master’s degree program directly. The requirements are the same as for the M.S. en route to the Ph.D. This program is normally completed in one year, but a part-time program may be spread over as many as four years.

A brochure providing additional information about the department, faculty, courses, and facilities is available from the Graduate Coordinator, Department of Computer Science, Yale University, PO Box 208285, New Haven CT 06520-8285; e-mail, cs-admissions@cs.yale.edu.

COURSES

CPSC 522a, Operating Systems Zhong Shao
The design and implementation of operating systems. Topics include synchronization, deadlocks, process management, storage management, file systems, security, protection, and networking.

CPSC 523b, Principles of Operating Systems Abraham Silberschatz
A survey of the underlying principles of modern operating systems. Topics include process management, memory management, storage management, protection and security, distributed systems, and virtual machines. Emphasis on fundamental concepts rather than implementation.

CPSC 524a, Parallel Programming Techniques Andrew Sherman
Practical introduction to parallel programming, emphasizing techniques and algorithms suitable for scientific and engineering computations. Aspects of processor and machine architecture. Techniques such as multithreading, message passing, and data parallel computing using graphics processing units. Performance measurement, tuning, and debugging of parallel programs. Parallel file systems and I/O.

CPSC 527a or b, Object-Oriented Programming Staff
Object-oriented programming as a means to efficient, reliable, modular, reusable code. Use of classes, derivation, templates, name-hiding, exceptions, polymorphic functions, and other features of C++.

CPSC 531a, Computer Music: Algorithmic and Heuristic Composition Scott Petersen
Study of the theoretical and practical fundamentals of computer-generated music. Music and sound representations, acoustics and sound synthesis, scales and tuning systems, algorithmic and heuristic composition, and programming languages for computer music. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language.

CPSC 532b, Computer Music: Sound Representation and Synthesis Scott Petersen
Study of the theoretical and practical fundamentals of computer-generated music, with a focus on low-level sound representation, acoustics and sound synthesis, scales and tuning systems, and programming languages for computer music generation. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language. Prerequisite: ability to read music.

CPSC 533a, Computer Networks Yang Yang
An introduction to the design, implementation, analysis, and evaluation of computer networks and their protocols. Topics include layered network architectures, applications, transport, congestion, routing, data link protocols, local area networks, performance analysis, multimedia networking, network security, and network management. Emphasis on protocols used in the Internet.

CPSC 534b, Topics in Networked Systems Yang Yang
Study of networked systems such as the Internet and mobile networks which provide the major infrastructure components of an information-based society. Topics include the design principles, implementation, and practical evaluation of such systems in new settings, including cloud computing, software-defined networking, 5G, Internet of things, and vehicular networking.

CPSC 537a, Introduction to Database Systems Abraham Silberschatz

CPSC 539b, Software Engineering Ruzica Piskac
Introduction to building a large software system in a team. Learning how to collect requirements and write a specification. Project planning and system design. Increasing software reliability: debugging, automatic test generation. Introduction to type systems, static analysis, and model checking.

CPSC 546b, Data and Information Visualization Holly Rushmeier
Visualization is a powerful tool for understanding data and concepts. This course provides an introduction to the concepts needed to build new visualization systems, rather than to use existing visualization software. Major topics are abstracting visualization tasks, using visual channels, spatial arrangements of data, navigation in visualization systems, using multiple views, and filtering and aggregating data. Case studies to be considered include a wide range of visualization types and applications in humanities, engineering, science, and social science. Prerequisite: CPSC 223.

CPSC 551b, The User Interface David Gelernter
The user interface (UI) in the context of modern design, where tech has been a strong and consistent influence from the Bauhaus and U.S. industrial design of the 1920s and 1930s through the IBM-Eames design project of the 1950s to 1970s. The UI in the context of the
windows-menu-mouse desktop, as developed by Alan Kay and Xerox in the 1970s and refined by Apple in the early 1980s. Students develop a detailed design and simple implementation for a UI.

**CPSC 553a / CB&B 555a / GENE 555a, Machine Learning for Biology**  
Smrita Krishnaswamy  
This course introduces biology as a systems and data science through open computational problems in biology, the types of high-throughput data that are being produced by modern biological technologies, and computational approaches that may be used to tackle such problems. We cover applications of machine-learning methods in the analysis of high-throughput biological data, especially focusing on genomic and proteomic data, including denoising data; nonlinear dimensionality reduction for visualization and progression analysis; unsupervised clustering; and information-theoretic analysis of gene regulatory and signaling networks. Students' grades are based on programming assignments, a midterm, a paper presentation, and a final project.

**CPSC 554a, Software Analysis and Verification**  
Ruzica Piskac  
Introduction to concepts, tools, and techniques used in the formal verification of software. State-of-the-art tools used for program verification; detailed insights into algorithms and paradigms on which those tools are based, including model checking, abstract interpretation, decision procedures, and SMT solvers.

**CPSC 556b / ENAS 951b, Wireless Technologies and the Internet of Things**  
Wenjun Hu  
Fundamental theory of wireless communications and its application explored against the backdrop of everyday wireless technologies such as WiFi and cellular networks. Channel fading, MIMO communication, space-time coding, opportunistic communication, OFDM and CDMA, and the evolution and improvement of technologies over time. Emphasis on the interplay between concepts and their implementation in real systems. The labs and homework assignments require Linux and MATLAB skills and simple statistical and matrix analysis (using built-in MATLAB functions).

**CPSC 565b, Theory of Distributed Systems**  
James Aspnes  
Models of asynchronous distributed computing systems. Fundamental concepts of concurrency and synchronization, communication, reliability, topological and geometric constraints, time and space complexity, and distributed algorithms.

**CPSC 567a, Cryptography and Computer Security**  
Staff  
A survey of such private and public key cryptographic techniques as DES, RSA, and zero-knowledge proofs, and their application to problems of maintaining privacy and security in computer networks. Focus on technology, with consideration of such societal issues as balancing individual privacy concerns against the needs of law enforcement, vulnerability of societal institutions to electronic attack, export regulations and international competitiveness, and development of secure information systems.

**CPSC 568a, Computational Complexity**  
Joan Feigenbaum  
Introduction to the theory of computational complexity. Basic complexity classes, including polynomial time, nondeterministic polynomial time, probabilistic polynomial time, polynomial space, logarithmic space, and nondeterministic logarithmic space. The roles of reductions, completeness, randomness, and interaction in the formal study of computation.

**CPSC 570b, Artificial Intelligence**  
Brian Scassellati  
Introduction to artificial intelligence research, focusing on reasoning and perception. Topics include knowledge representation, predicate calculus, temporal reasoning, vision, robotics, planning, and learning.

**CPSC 573a, Intelligent Robotics Laboratory**  
Brian Scassellati  
Students work in small teams to construct novel research projects using one of a variety of robot architectures. Project topics may include human-robot interaction, adaptive intelligent behavior, active perception, humanoid robotics, and socially assistive robotics.

**CPSC 574a, Computational Intelligence for Games**  
James Glenn  
An overview of computational vision with a biological emphasis. Suitable as an introduction to biological perception for computer science and engineering students, as well as an introduction to computational vision for mathematics, psychology, and physiology students.

**CPSC 575a / ENAS 575a, Computational Vision and Biological Perception**  
Steven Zucker  
Advanced view of vision from a mathematical, computational, and neurophysiological perspective. Emphasis on differential geometry, machine learning, visual psychophysics, and advanced neurophysiology. Topics include perceptual organization, shading, color, and texture.

**CPSC 576b / AMTH 667b / ENAS 576b, Advanced Computational Vision**  
Steven Zucker  
Introduction to basic concepts of two- and three-dimensional computer graphics. Topics include affine and projective transformations, clipping and windowing, visual perception, scene modeling and animation, algorithms for visible surface determination, reflection models, illumination algorithms, and color theory.
CPSC 579b, Advanced Topics in Computer Graphics  Julie Dorsey
An in-depth study of advanced algorithms and systems for rendering, modeling, and animation in computer graphics. Topics vary and may include reflectance modeling, global illumination, subdivision surfaces, NURBS, physically based fluids systems, and character animation.

CPSC 635b, Topics on the Hardware/Software Interface  Abhishek Bhattacharjee
This course focuses on advanced topics in computer systems, particularly at the intersection of architecture and systems software (i.e., operating systems, firmware, device drivers, etc.). The goal is to give students exposure to building hardware and low-level software support for emerging high-performance server systems used in data centers. Key topics include the virtual memory abstraction, cache coherence, and memory consistency, particularly in the context of performance and energy efficiency. We study the impact of hardware heterogeneity on these trends; and the emergence of hardware accelerators (i.e., GPUs, FPGAs, fixed-function accelerators, etc.) and novel memory technologies (i.e., high-bandwidth die-stacked memory) introduces many system performance and programmability challenges. The course prepares students to understand these challenges and provides the background to architecture systems that embrace extreme heterogeneity. Prerequisites: the course is aimed at Computer Science graduate students with a background in systems programming; solid undergraduate exposure to systems programming (CPSC 323) and operating systems (CPSC 422 and/or CPSC 423) is assumed; exposure to other systems classes (e.g., CPSC 424, CPSC 437) is useful but not required. Students who do not fit this profile may be allowed to enroll with permission of the instructor.

CPSC 640b, Topics in Numerical Computation  Vladimir Rokhlin
This course discusses several areas of numerical computing that often cause difficulties to non-numericists, from the ever-present issue of condition numbers and ill-posedness to the algorithms of numerical linear algebra to the reliability of numerical software. The course also provides a brief introduction to “fast” algorithms and their interactions with modern hardware environments. The course is addressed to Computer Science graduate students who do not necessarily specialize in numerical computation; it assumes the understanding of calculus and linear algebra and familiarity with (or willingness to learn) either C or FORTRAN. Its purpose is to prepare students for using elementary numerical techniques when and if the need arises.

CPSC 659a, Building Interactive Machines  Marynel Vazquez
This course brings together methods from machine learning, computer vision, robotics, and human-computer interaction to enable interactive machines to perceive and act in dynamic environments. Part of the course examines approaches for perception with a variety of devices and algorithms; the other part focuses on methods for decision-making. The course is a combination of lectures, reviews of state-of-the-art papers, discussions, coding homework, and a final team project. Prerequisites: a basic understanding of probability, calculus, and algorithms is expected, as well as proficiency in Python and high-level familiarity with C++. Students who do not fit this profile may be allowed to enroll with permission of the instructor.

CPSC 662a / AMTH 561a, Spectral Graph Theory  Daniel Spielman
An applied approach to spectral graph theory. The combinatorial meaning of the eigenvalues and eigenvectors of matrices associated with graphs. Applications to optimization, numerical linear algebra, error-correcting codes, computational biology, and the discovery of graph structure.

CPSC 663b / AMTH 663b, Deep Learning Theory and Applications  Smita Krishnaswamy
Deep neural networks have gained immense popularity in the past decade due to their outstanding success in many important machine-learning tasks such as image recognition, speech recognition, and natural language processing. This course provides a principled and hands-on approach to deep learning with neural networks. Students master the principles and practices underlying neural networks, including modern methods of deep learning, and apply deep learning methods to real-world problems including image recognition, natural language processing, and biomedical applications. Course work includes homework and a final project – either group or individual, depending on the total number enrolled—with both a written and oral (i.e., presentation) component.

CPSC 677a, Advanced Natural Language Processing  Dragomir Radev
Advanced topics in natural language processing (NLP), including related topics such as deep learning and information retrieval. Included are: (1) fundamental material not covered in the introductory NLP class such as text summarization, question answering, document indexing and retrieval, query expansion, graph-based techniques for NLP and IR, as well as (2) state-of-the-art material published in the past few years such as transfer learning, generative adversarial networks, reinforcement learning for NLP, sentence representations, capsule networks, multitask learning, and zero-shot learning. Prerequisite: CPSC 570, CPSC 577, or equivalent, or permission of the instructor.

CPSC 678b, Creative Artificial Intelligence for Visual Computing  Julie Dorsey
How can artificial intelligence help us create visual content? In this readings-and-projects-based course, we explore how to use tools such as probabilistic models, probabilistic programs, and neural networks to generate content, explore design spaces, and support creativity for 2D and 3D graphics and vision applications. Each week, we read recent papers from the visual computing and AI literatures and discuss their contributions, connections, and limitations. Students also complete a collaborative, open-ended final project. Throughout, the course emphasizes key academic skills such as critical paper-reading and how to give clear and compelling presentations. Topics include arranging objects, controlling procedural models, generating 3D geometry, writing and drawing, assigning materials and colors, generative adversarial models, learning representations of shape, learning from RGB-D panoramas, deep reinforcement learning, and exploring manifolds. Prerequisites: equivalent of CPSC 323 and some background in computer graphics, artificial intelligence, machine learning, or probabilistic modeling are helpful. Students are not expected to be familiar with all of these areas; this is a multidisciplinary
area, and we welcome students of diverse backgrounds to share their expertise and interests. Students who lack this background may be allowed to enroll with permission of the instructor.

**CPSC 690a or b, Independent Project I**  Staff
By arrangement with faculty.

**CPSC 691a or b, Independent Project II**  Staff
By arrangement with faculty.

**CPSC 692a or b, Independent Project**  Staff
Individual research for students in the M.S. program. Requires a faculty supervisor and the permission of the director of graduate studies.

**CPSC 752b / CB&B 752b / MB&B 752b / MCDB 752b, Biomedical Data Science: Mining and Modeling**  Mark Gerstein
Biomedical data science encompasses the analysis of gene sequences, macromolecular structures, and functional genomics data on a large scale. It represents a major practical application for modern techniques in data mining and simulation. Specific topics to be covered include sequence alignment, large-scale processing, next-generation sequencing data, comparative genomics, phylogenetics, biological database design, geometric analysis of protein structure, molecular-dynamics simulation, biological networks, normalization of microarray data, mining of functional genomics data sets, and machine-learning approaches to data integration. Prerequisites: biochemistry and calculus, or permission of the instructor.

**CPSC 800a or b, Directed Readings**  Staff
By arrangement with faculty.

**CPSC 990a, Ethical Conduct of Research for Master's Students**  Holly Rushmeier
This course meets on four consecutive Fridays.

**CPSC 991a / MATH 991a, Ethical Conduct of Research**  Alexander Goncharov
Course end